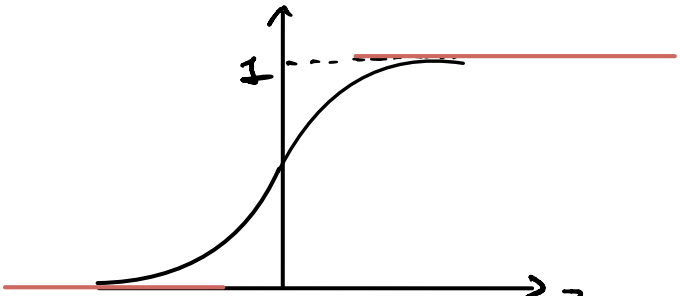
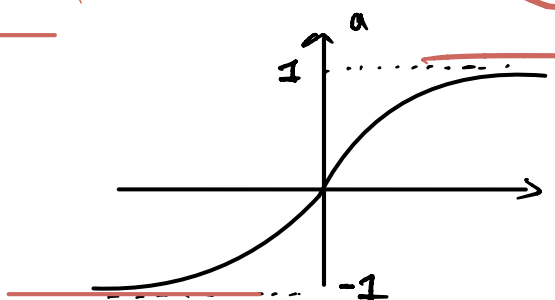
Activation Functions

Slopes at these points are close to zero, which can slow down gradient descent.



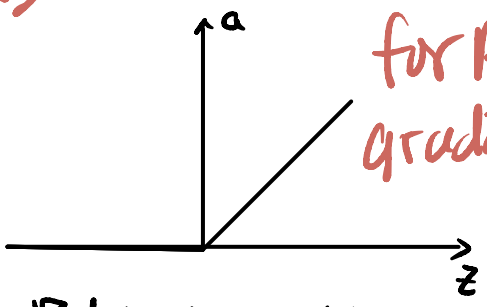
sigmoid: $a = \frac{1}{1 + e^{-z}}$



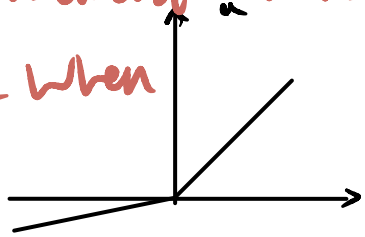
tanh: $\frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$

⇒ Solution: Relu and leaky Relu functions:

for Relu, gradient = 0 when $z \leq 0$; and gradient = 1 when $z > 0$



Relu $a = \max(0, z)$



can be set as a hyperparameter.

leaky Relu $a = \max(0.01z, z)$

Why do you need non-linear Activation functions?

Counter example: only use linear Activation functions.

Given x:

$z^{[1]} = W^{[1]}x + b^{[1]}$

$a^{[1]} = z^{[1]}$ (identity)

$g(z) = z$: linear activation function

no use of non-linear activation function here.

$z^{[2]} = W^{[2]}a^{[1]} + b^{[2]}$

$a^{[2]} = z^{[2]}$

$$\rightarrow a^{[1]} = z^{[1]} = w^{[1]}x + b^{[1]}$$

$$a^{[2]} = z^{[2]} = w^{[2]}a^{[1]} + b^{[2]}$$

$$= (w^{[1]}x + b^{[1]})w^{[2]} + b^{[2]}$$

$$= \underbrace{(w^{[1]}w^{[2]})}_{w'}x + \underbrace{(b^{[1]}w^{[2]} + b^{[2]})}_{b'}$$

$$= w'x + b'$$ still a linear function

→ the hidden layers of the neural net are redundant.

Derivatives of non-linear activation functions.

Sigmoid.

$$g(z) = \frac{1}{1+e^{-z}} \qquad g'(z) = \underbrace{a}_{\text{activation}}(1-a)$$

$a = g(z)$

Tanh:

$$g(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \qquad g'(z) = 1 - a^2$$

Relu:

$$g(z) = \max(0, z) \qquad g'(z) = \begin{cases} 0 & \text{if } z < 0 \\ 1 & \text{if } z \geq 0 \end{cases}$$

Although $g(z)$ at $z=0$ is actually not defined.

Leaky Relu:

$$g(z) = \max(0.01z, z). \qquad g'(z) = \begin{cases} 0.01 & \text{if } x < 0 \\ 1 & \text{if } x > 0 \end{cases}$$

Tanh is universally better than sigmoid, except the case where we are dealing a regression problem. Relu and leaky Relu, on the other hand, are almost always more efficient. Relu is most often set as default activation function